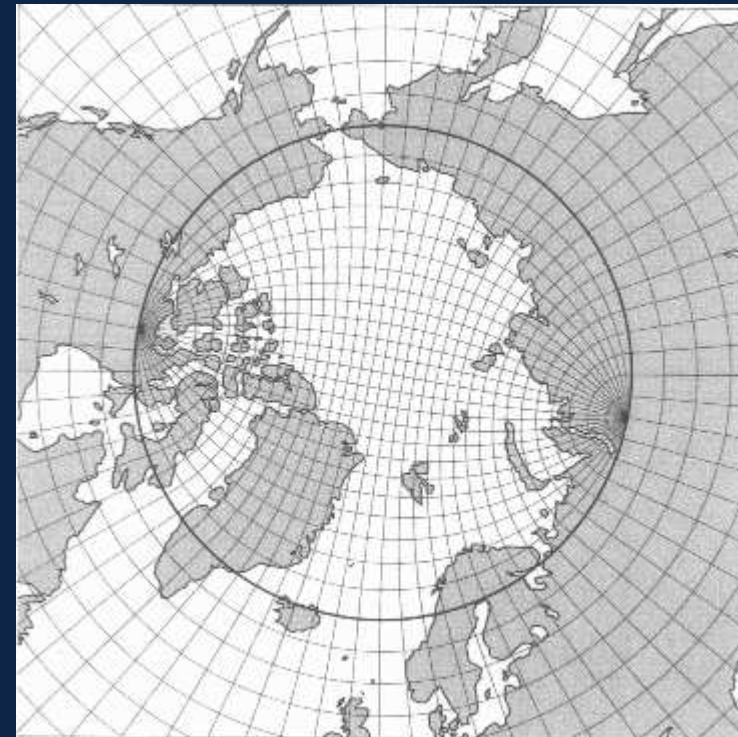


Fast regridding of complex grids for visualization

Jon Blower, University of Reading, UK
j.d.blower@reading.ac.uk

Motivation

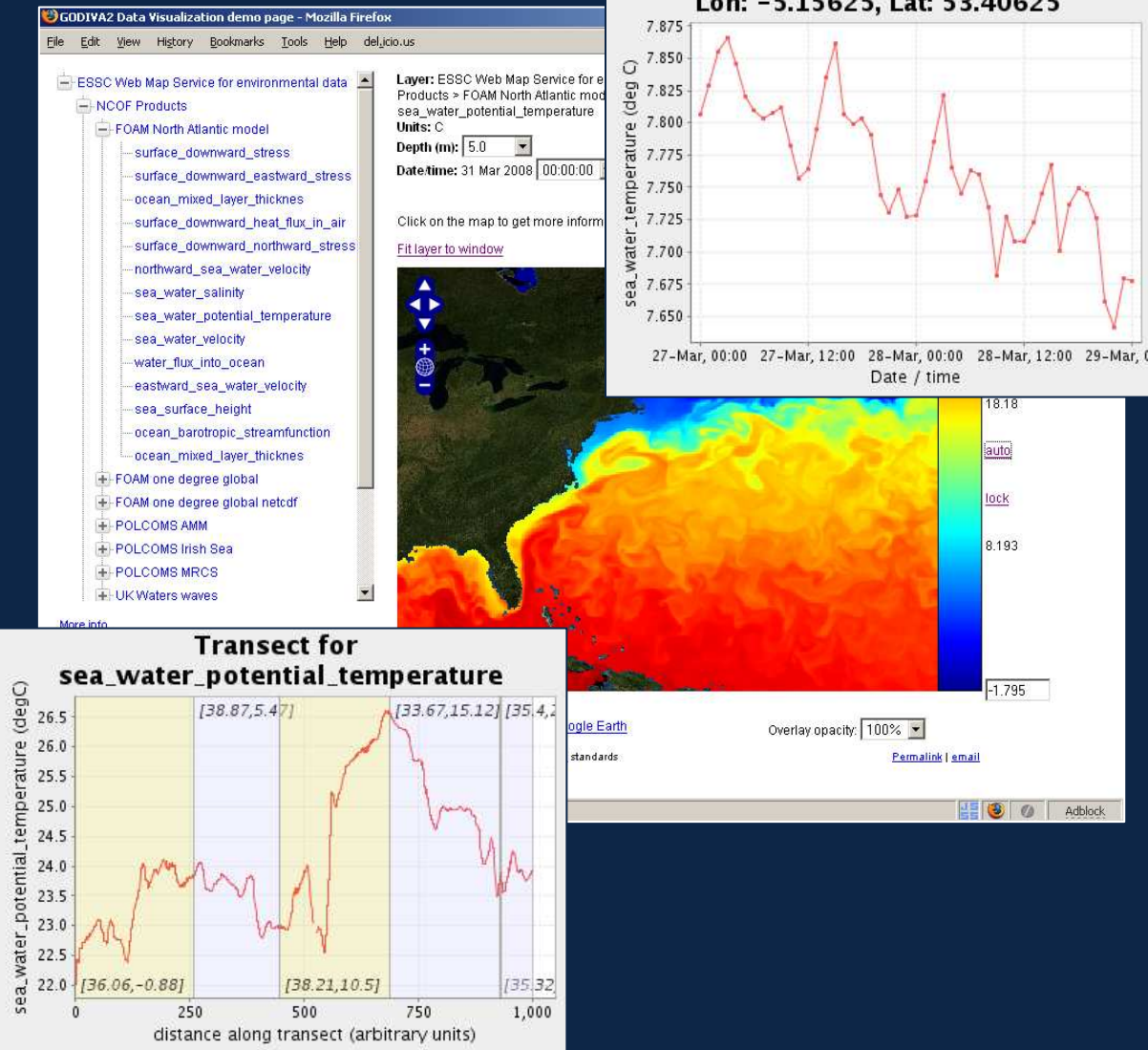
- Complex grids becoming more common
 - Following complex boundaries (e.g. coasts)
 - Avoiding polar singularities
 - Maintaining constant spatial resolution
 - Satellite swaths
- We focus here on rectangular, distorted grids
 - Not cube-sphere, yin-yang etc (yet)
- CF defines how to encode these
 - “two-dimensional coordinate axes”
 - Or “curvilinear grids”



We need this for ncWMS (and THREDDS-WMS)

<http://www.reading.ac.uk/godiva2>

- To create maps
- To extract information at a given point
 - Timeseries
 - Profiles
- To produce transects
- And we need to do this **quickly**
 - We want a 256x256 map image in less than a second
- And **on the fly**



In other words:

- We need to find the nearest i,j index in the grid
- For any combination of latitude/longitude

- CF defines this:

```
LatLon ll = grid.getLatLon(GridPoint gp);
```

- We need the inverse:

```
GridPoint gp = grid.getNearest(LatLon ll);
```

- But most curvilinear grids are not analytically invertible!

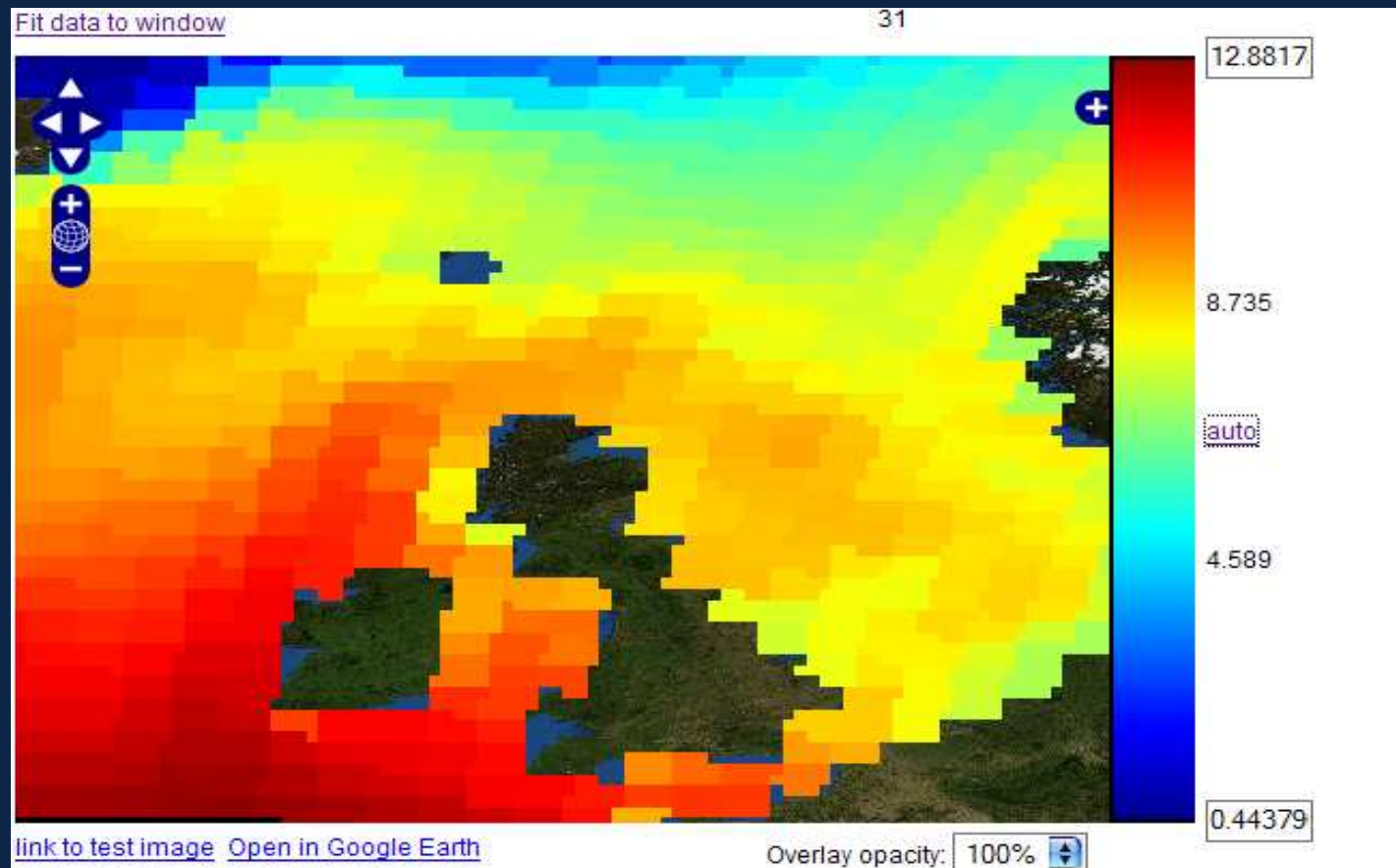
Attempt 1: Look-Up Tables

- `getNearest()` can be very slow in the general case
 - Exhaustive search
- So we calculated nearest grid point for a fixed set of lon/lat pairs
 - This is a look-up table (LUT)
 - Calculated offline in a slow process
 - Then loaded into memory
- (Aside: we used Rtrees to speed up the generation of the LUT from hours to minutes.)

Results:

Not bad, could do better

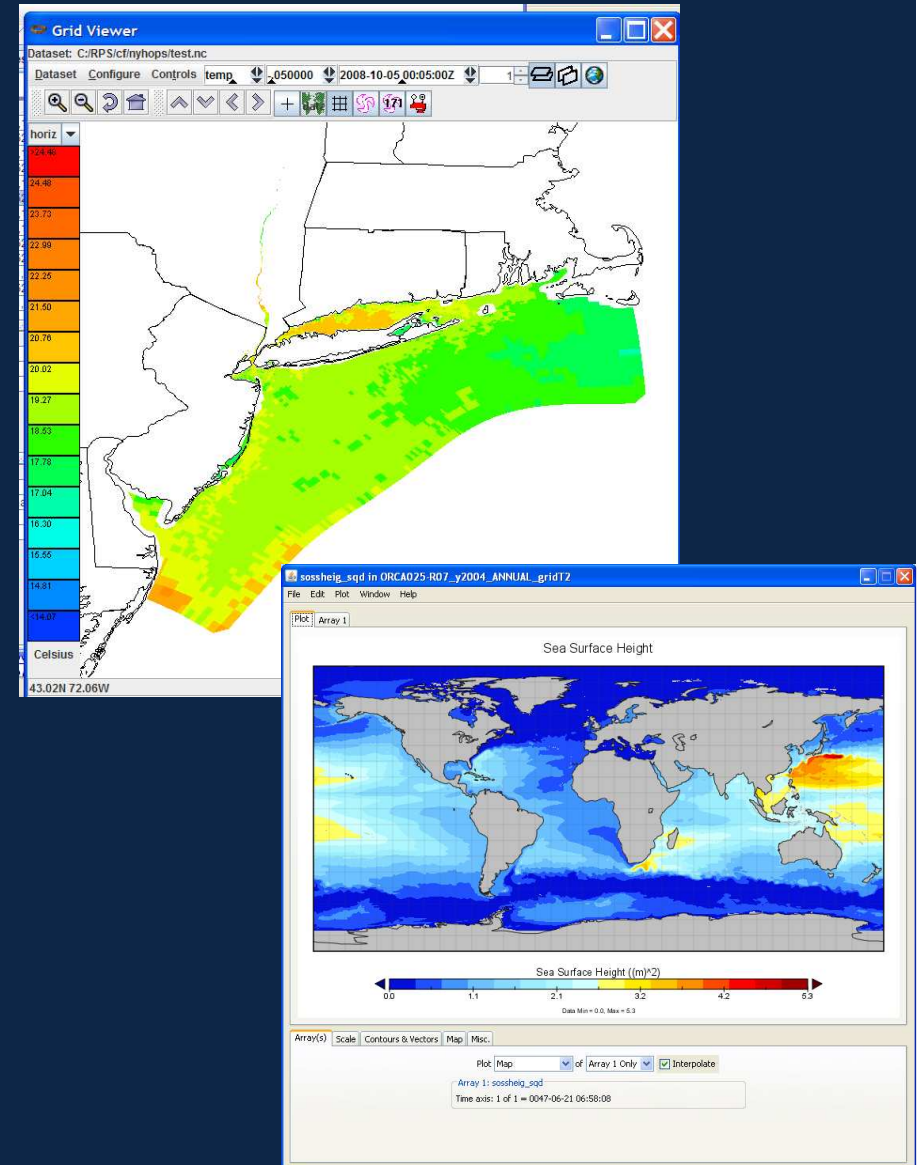
NEMO
tripolar
grid



- Very fast at generating images
- But cells appear blocky at high zooms
- (Also, configuration is awkward)

An alternative approach

- ToolsUI and Panoply use a different method
- They cycle through each grid cell and calculate its bounding polygon
- Then paint the polygon onto a canvas
- (They calculate polygons in different ways... ToolsUI is correct I think.)



Why can't we simply use the ToolsUI/Panoply method?

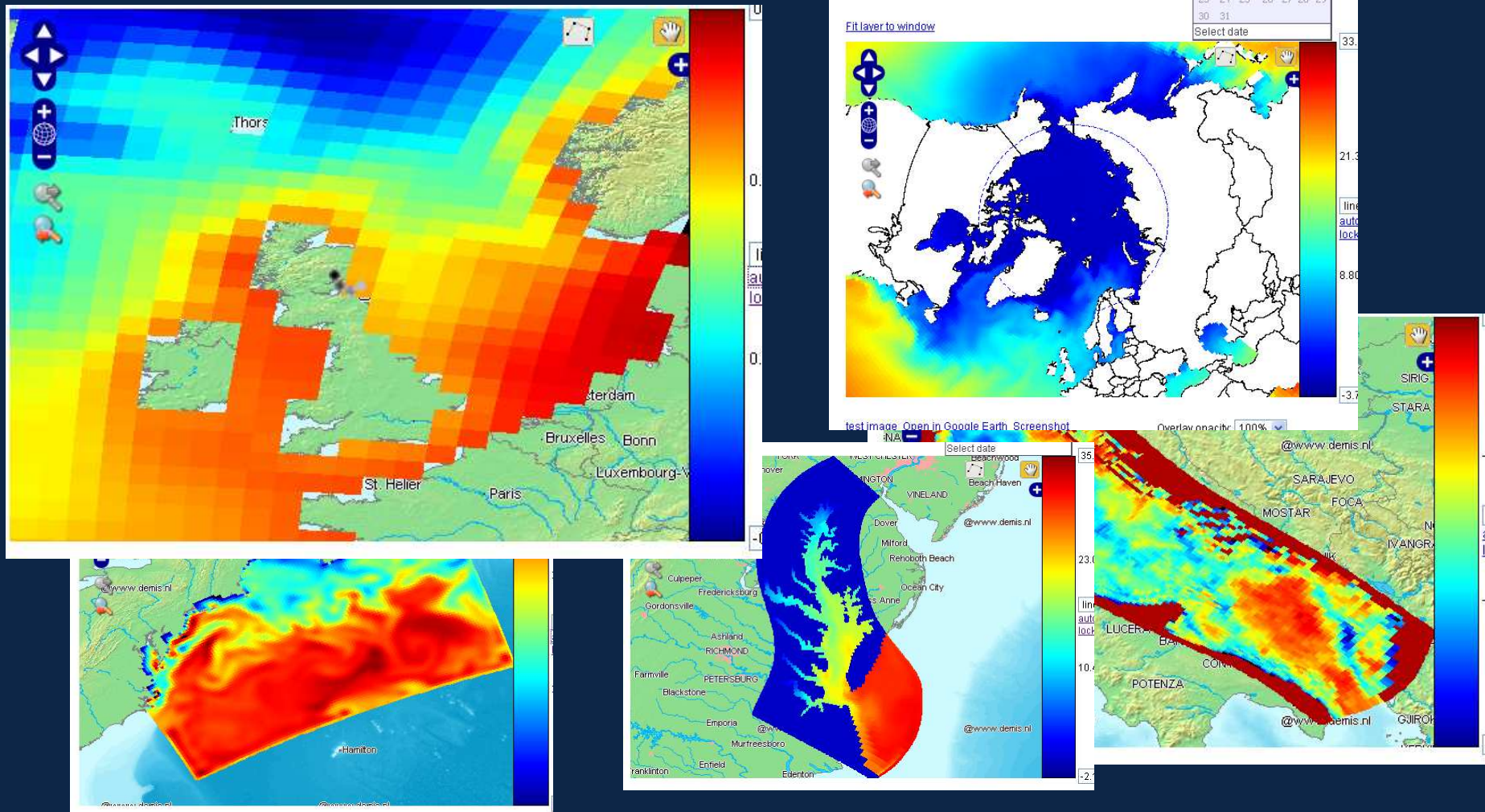
- We need the `getNearest()` function for things other than generating map images
 - Timeseries, profiles, transects...
- Generation of map images still not fast enough
 - Takes several seconds, we need $<1s$
 - Inefficient when under- or over-sampling
- Difficulties when creating images in other projections
 - E.g. polar stereographic
- (Quite a few other reasons too... no time today!)

A hybrid solution

- We use the ToolsUI/Panoply method to generate LUTs
 - Takes seconds, not minutes/hours
 - (some cunning use of Java image APIs here...)
- The LUT gets us close to the correct grid point
 - Behaves like a fast spatial index
- Then we exhaustively search the nearby grid points for a better match

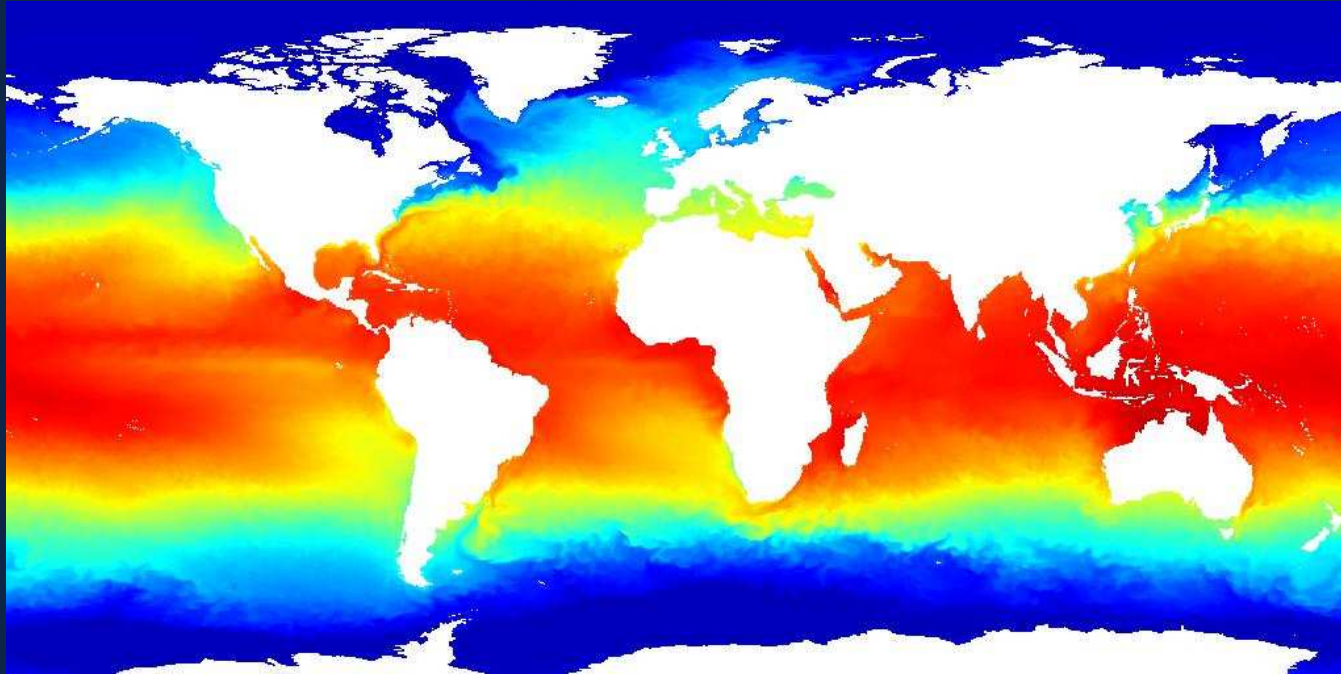
It works! (most of the time)

<http://www.reading.ac.uk/godiva2>



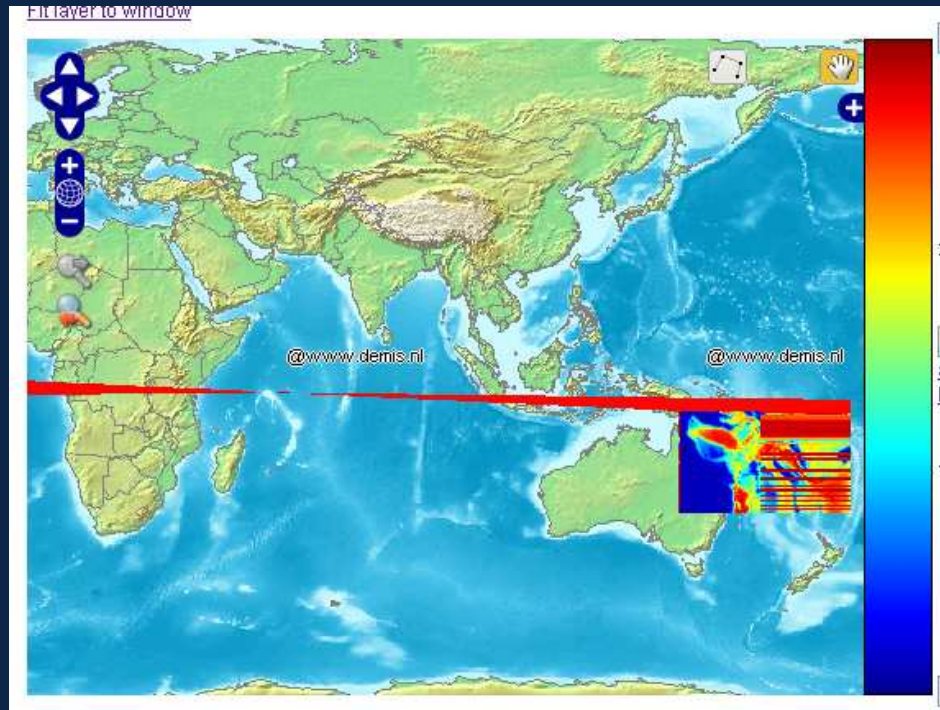
- (problems with land masks are in the data, not the software ;-)

Larger-scale regridding...



- Source data: $\frac{1}{4}$ degree global, NEMO tripolar grid
 - (1 million grid points in the horizontal)
- Final image: 1024 x 512
- Time taken: ~7 seconds

Rarely, it screws up... ☹



- Problems can occur when it's hard to calculate the bounding polygon of a grid cell
 - Some curvilinear grid formulations are “badly-behaved”
 - Wrapping around the anti-meridian can cause problems

Suggestions for CF conventions

- Encode boundaries of grid cells as polygons
 - These can be hard to calculate if only the centres are known
 - Maybe following ADAGUC (see John van de Vegte's talk)
 - (note added after presentation: already exists in CF version 1.4, section 7.1. But nobody seems to use it!)
- Encode look-up table (rough mapping of lat/lon to i/j) in headers
 - Would help simple viz clients to display data
- Both could be optional

Conclusions

- We have reusable Java code for regridding curvilinear grids into any other projection
 - Great for GIS integration!
 - Nearest-neighbour interpolation only
 - Regridding is fast (seconds or less, not minutes or hours)
- Designed mainly for visualization but could be used for data analysis too
 - May require closer scrutiny
- Part of ncWMS codebase (ncwms.sf.net), but will be factored out into separate library

THANKS!

Contact: j.d.blower@reading.ac.uk

**Thanks especially to Rich Signell
for providing suitably-awkward
datasets for test purposes ;-)**